

REMARKS

The Office Action mailed March 22, 2007 has been carefully considered. In this non-final Office Action, Claims 1-8 are pending. Claims 1-8 are rejected. No claims have been amended in this Office Action. The title was rejected as non descriptive and a new title was requested.

Examiner rejected Claims 1-8 under 35 U.S.C. § 102(e) as being un-patentable over U.S. Patent No. 6,073,218 to DeKoning et al. ("DeKoning"). Applicants respectfully argue that this reference does not teach the current invention. These grounds for rejection are hereby traversed and allowance of the claims is requested in view of the following arguments.

The Office Action stated the title was not descriptive and Applicants have changed the title. The new title is "A method for automatically configuring a balanced mass storage system for measuring system performance."

Applicants' invention, in general, relates to a method and system for automatically configuring a mass storage system for measuring system performance. The system comprises a plurality of disk storage elements, each element having at least one hyper and connected to a disk storage controller. The disk storage controller is connected to at least one host computer. The host computer defines a plurality of logical units (LUNs) and the storage controller defines in its configuration a front-end hierarchy and a back-end hierarchy. Where the balancing assignments of LUNS across the back-end and marginally balancing assignments of disk drive elements and hypers of the disk drive elements across the back-end hierarchy.

35 USC § 102 Rejections

Grounds for reconsideration of Claim 1, 3, 5, 7

Claims 1, 3, 5, and 7 were rejected based on DeKoning. As Examiner has rejected these claims together, for the sake of brevity, Applicants respond to these rejections in one argument.

Applicants' Claim 1 states:

A method for automatically configuring a mass storage system for measuring system performance, the system comprising a plurality of disk storage elements, each element having at least one hyper and connected to a disk storage controller, the disk storage controller being connected to at least one host computer, the host computer defining a plurality of logical units (LUN's), the storage controller defining in its configuration a front- end hierarchy and a back-end hierarchy, the method comprising

balancing assignments of LUN's across the back-end hierarchy, and

marginally balancing assignments of disk drive elements and hypers of the disk drive elements across the back-end hierarchy.

Claim 1 is similar in scope to Claim 3. Claim 3 differs from Claim 1 by replacing "balancing assignments of LUNs across the back-end hierarchy, and marginally balancing assignments of disk drive elements and hypers of the disk drive elements across the back-end hierarchy" with the following claim language:

"balancing assignments of components of the back-end hierarchy to LUNs of the front-end hierarchy , and marginally balancing said assignment."

Claims 5 and 7 are program product claims corresponding to Claim 1 and 3, respectively. As the Office Action responds to these with one rejection, Applicants respond in kind with one argument.

Examiner asserts that DeKoning discloses a method for automatically configuring a mass storage system for measuring system performance, the system comprising a plurality of disk storage elements, each element having at least one hyper and connected to a disk storage controller, the disk storage controller being connected to at least one host computer, the host computer defining a plurality of logical units numbers (sic), the storage controller defining in its configuration a front-end and a back-end hierarchy. The Examiner further asserts that DeKoning discloses the step of balancing assignment of LUNs across the back-end hierarch and marginally balancing assignments of disk drive elements across the back-end hierarchy. Applicants respectfully disagree.

DeKoning teaches “methods and associated apparatus which permit all of a plurality of storage controllers to share access to common storage devices of a storage subsystem.” (Column 3 lines 8-9). “In particular, [DeKoning] provides for concurrent processing by a plurality of RAID controllers.” (Column 3 lines 10-11). Per DeKoning, the acronym RAID stands for “Redundant Arrays of Inexpensive [independent] Disks” and is a system for “increased capacity and reliability” in mass storage systems. (Column 1, lines 40-41, lines 53-55).

The “RAID storage systems typically utilize a control module,” and the [RAID] “(sub)system appears[s] to the host computer as a single . . . disk drive.” (Column 1 line 65-Column 2 line 3). In an effort “to further improve reliability,” RAID systems “provide redundant control modules.” (Column 2 lines 15-16). DeKoning also states that in RAID

systems “any number of control modules may participate in the redundant processing . . . however [in his definition of the prior art] when all redundant control modules are operable a significant portion of the processing power of the redundant control modules is wasted.” (Column 2 lines 31-38). The DeKoning “invention solves the above [wasted processing power] and other problems.” (Column 3 lines 5-7).

DeKoning does this by using a method that “provides for concurrent processing by a plurality of RAID controllers.” (Column 3 lines 10-11). That is, DeKoning provides a system where “the primary controller is responsible for fairly sharing access to the common disk drives of the LUN among all requesting RAID controllers.” (Column 3 lines 61-65). In “fairly sharing access,” DeKoning does not provide a “method for automatically configuring a mass storage system for measuring system performance.” Rather, DeKoning manages a system to improve the efficiency “processing power,” and his system requires access in order to function.

DeKoning functions by having “the primary controller . . . fairly sharing access to the common disk drives of the LUNs.” “A controller desiring access to the shared disk drives of the LUN sends a message to the primary controller.” (Column 3 lines 62-65). “The primary controller returns a grant of the requested lock in due course when such exclusively is permissible.” (Column 3 line 65-Column 4 line 2). The primary controller does this through the use of “semaphores for all controllers accessing the shared LUNs.” (Column 4 lines 5-8). This requires that there be access requests for DeKoning to function. Further, DeKoning states his invention is responding to an identified problem in the prior art that “a significant portion of the processing power of the redundant control modules is wasted.”

This highlights two ways in which DeKoning is different than the current invention which provides “a method for automatically configuring a mass storage system for measuring system performance . . . the method comprising balancing assignments of LUN’s across the back-end hierarchy, and marginally balancing assignments of disk drive elements and hypers of the disk drive elements across the back-end hierarchy.” First, DeKoning states he fixes an identified problem. On the other hand, the current invention provides a “method for automatically configuring a mass storage system for measuring system performance,” and this can be used to find problems. Second, DeKoning’s method, “sharing access to common drives” requires access for it to function e.g. “each semaphore may be dynamically assigned by the primary control module.” (Column 10 lines 10-12). In requiring access, access requests must first be generated for the controllers to actively request the “locks.” Without access requests controllers do not request locks, and without locks DeKoning does not function. Conversely, the current invention contains no such dynamic limitation and can be performed statically. Therefore DeKoning is not equivalent to a “method for automatically configuring a mass storage system for measuring system performance.”

Balancing controller access to particular LUNs, DeKoning also does not balance the back end of the system. DeKoning functions by “utilizing otherwise idle processing power of the additional [redundant] . . . control modules to process additional host generated I/O requests.” (Column 9 lines 25-28). This means that “there may be any number of active controllers with respect to any particular LUN.” (Column 9 lines 29-31). Where “a plurality of active controllers may be processing I/O requests for particular shared LUN in parallel.” (Column 9 lines 37-39). The LUNs are on the front end of the system and DeKoning doesn’t balance access just controls

access to maintain drive consistency based on access locks assigned to LUNs or the striping of the LUNs. Therefore, not considering elements on the back end of the system, DeKoning does not balance using components from the back end of the system, does not address the units on the back end of the system, and does not disclose “balancing . . . across the back-end hierarchy.”

Based on the foregoing arguments, Applicants respectfully request reconsideration and removal of the cited 35 U.S.C. § 102(e) rejection for Claims 1, 3, 5, and 7. As Claims 2, 4, 6, and 8 are dependant on claims now believed allowable, Applicants also request reconsideration and removal of the rejections for Claims 2, 4, 6, and 8.

Grounds for reconsideration of Claim 2, 4, 6, 8

With respect to claims 2, 4, 6, and 8 Applicants incorporate the aforementioned arguments and further argue that DeKoning does not disclose wherein the balancing of assignments across the back-end hierarchy maintains the assignment at any level of the back-end hierarchy within one assignment value. As stated above, DeKoning does not consider “balancing assignment of LUN’s across the back-end hierarchy, and marginally balancing assignments of disk drive elements and hypers of the disk drive elements across the back-end hierarchy.” Rather, DeKoning uses the multiple controller modules in a system to increase system throughput. DeKoning also addresses the problem caused by more than one controller module concurrently requesting access to a drive through the use of locking semaphores.

As DeKoning does not balance the same type of devices it can not “maintain the assignment at any level of the back-end hierarchy within one assignment value.” Further, the use of semaphores to lock stripes on the LUNs, does not function by balancing the LUNs or maintaining the assignment level . . . within one assignment value.” Semaphores, as used in

DeKoning, are to “grant temporary exclusive access” to a portion of the LUN. (Column 11, lines 62-63).

DeKoning functions by having “the primary controller allocate[s] a pool of semaphore tables.” (Column 11 lines 63-64). “Each [semaphore] table identifies the LUN to which it corresponds.” (Column 12, lines 1-2). “Each semaphore corresponds to a temporary lock over a particular region of the associated LUN by a particular control module.” (Column 12, lines 23-25). “The maximum number of semaphore entries in the semaphore list of a semaphore table depends on the maximum number of regions that may be simultaneously locked.” (Column 12, lines 43-45).

When a “control module desires temporary access exclusive access to the disk array . . . access is requested by a locking semaphore.” (Column 9 lines 56-58). “If the semaphore is successfully locked, the requesting control module may proceed.” (Column 9, 59-61). However, “until the semaphore is successfully locked, the requesting control module is suspended with respect to its processing.” (Column 9 lines 63-64). That is, “the newly requested lock is deferred until such a time as it may be granted.” (Column 10, lines 22-24). DeKoning does not state that there is a limit of the number of modules that could be suspended waiting for a semaphore lock, rather it seems any number of modules could be suspended waiting for a particular semaphore.

This methodology does not ensure that “the assignment at any level of the back-end hierarchy [is] within one assignment value.” DeKoning is not addressing the hierarchy or assignment of the hierarchy of the back-end of the system, rather DeKoning is controlling access to LUNs or the stripes located on LUNs. DeKoning provides a method to enable multiple

Applicant: William Glynn
U.S.S.N.: 10/821,696
Filing Date: April 9, 2004
EMC Docket No.: EMC-99-173CON1

controllers to access the LUNs as well as providing a routine to handle consistency or overlapping requests by controllers to the same portion of the LUN. This managing of controller access based on semaphores not the same where “balancing assignments across the back-end hierarchy maintains the assignment at any level of the back-end hierarchy within one assignment value.” Therefore, DeKoning does not teach “maintain[ing] the assignment at any level of the back-end hierarchy within one assignment value” and therefore can not properly be used for a 35 U.S.C. § 102(e) rejection. Based on the aforementioned arguments, Applicants respectfully request favorable reconsideration of the claims and removal of the 35 U.S.C. § 102(e) for Claims 2, 4, 6, and 8.

Applicant: William Glynn
U.S.S.N.: 10/821,696
Filing Date: April 9, 2004
EMC Docket No.: EMC-99-173CON1

Conclusion

In view of the foregoing, the Applicants believe that the application is in condition for allowance and respectfully request favorable reconsideration.

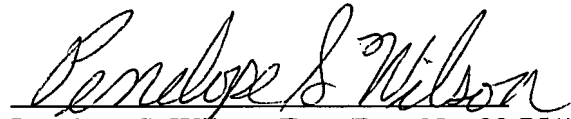
In the event the Examiner deems personal contact desirable in the disposition of this case, the Examiner is invited to call the undersigned attorney at (508) 293-7998.

Please charge all fees occasioned by this submission to Deposit Account No. 05-0889.

Respectfully submitted,

Dated: _____

5/15/07



Penelope S. Wilson, Esq. (Reg. No. 29,751)
Attorney for Applicants
EMC Corporation
Office of General Counsel
176 South Street
Hopkinton, MA 01748
Telephone: (508) 293-7998
Facsimile: (508) 293-7189